



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

3

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/657,989	09/09/2003	Shree A. Dandekar	DC-05164	1287

33438 7590 04/09/2007  
HAMILTON & TERRILE, LLP  
P.O. BOX 203518  
AUSTIN, TX 78720

EXAMINER

CHEN, QING

ART UNIT PAPER NUMBER

2191

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/09/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/657,989	DANDEKAR ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 06 February 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1,3-8,10-15 and 17-20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,3-8,10-15 and 17-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 06 February 2007 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                     | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

### DETAILED ACTION

1. This Office action is in response to the amendment filed on February 6, 2007.
2. **Claims 1, 3-8, 10-15, and 17-20** are pending.
3. **Claims 1, 3, 6-8, 10, 12-15, 17, and 20** have been amended.
4. **Claims 2, 9, and 16** have been cancelled.
5. The objections to the drawings are withdrawn in view of Applicant's amendments to the drawings and specification.
6. The objection to the abstract is withdrawn in view of Applicant's amendments to the abstract.
7. The objections to the specification are withdrawn in view of Applicant's amendments to the specification.
8. The objections to Claims 1, 3, 6-8, 10, 13, 15, 17, and 20 are withdrawn in view of Applicant's amendments to the claims.
9. The provisional obviousness-type double patenting rejections of Claims 1, 3, 5-8, 10, 12-15, 17, 19, and 20 are maintained in view of Applicant's arguments and further explained below. The provisional obviousness-type double patenting rejections of Claims 2, 9, and 16 are withdrawn in view of Applicant's cancellation of the claims.
10. The 35 U.S.C. § 112, second paragraph, rejections of Claims 1, 3-8, 10-15, and 17-20 are withdrawn in view of Applicant's amendments to the claims. The 35 U.S.C. § 112, second paragraph, rejections of Claims 2, 9, and 16 are withdrawn in view of Applicant's cancellation of the claims.

Art Unit: 2191

11. The 35 U.S.C. § 101 rejection of Claim 12 is withdrawn in view of Applicant's amendments to the claims.

### ***Response to Amendment***

#### ***Claim Objections***

12. **Claims 3, 10, and 17** are objected to because of the following informalities:

- **Claim 3** contains a typographical error: Claim 3 should depend on Claim 1, not Claim 2.
- **Claim 10** contains a typographical error: Claim 10 should depend on Claim 8, not Claim 9.
- **Claim 17** contains a typographical error: Claim 17 should depend on Claim 15, not Claim 16.

Appropriate correction is required.

#### ***Double Patenting***

13. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting

Art Unit: 2191

ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

14. **Claims 1, 3, 5-8, 10, 12-15, 17, 19, and 20** are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over Claims 1, 4-9, 12-17, and 20-24 of copending Application No. 10/768,823 (hereinafter referred to as Application '823) in view of Feinman (US 6,075,943) and further in view of Amberg et al. (US 5,991,543).

Claim 1 of Application '823 is compared to Claim 1 of the instant application in the table below. The only additions to the claim in the instant application are a script validation server that generates commands to automatically control the downloading of software images of the software file to a target information handling system and that the software file is transferred to a target information handling system after verification that the software file complies with a set of predetermined parameters and a compliance server to perform compliance verification to confirm that said software file complies with a predetermined set of rules.

In the same field of endeavor, Feinman discloses a system and method for remotely transferring and installing client server application programs from a source computer onto a remote client within a data processing system, where a sequential file—located on a server—is used to store information to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point, and the time that each application program is to be delivered (*see Figure 1B: 11; Figure 7: 100; Column 3: 3-5 and 44-67 through Column 4: 1-4*).

In the same field of endeavor, Amberg et al. disclose a method for sequencing software installation and/or testing steps for a computer system, where a sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record have corresponding entries and corresponding integers specified by CompID in a Component table (*see Figure 1: 100; Column 9: 9-16*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate a script validation server that generates commands to automatically control the downloading of software images of the software file to a target information handling system and a compliance server to perform compliance verification to confirm that said software file complies with a predetermined set of rules in the system of Application '823, since the system of Application '823 utilizes a compliance server and a download server to transfer the software file to a target information handling system. One would have been motivated to incorporate a script validation server that generates commands to automatically control the downloading of software images of the software file to a target information handling system, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Feinman – Column 1: 23-32*). One would also have been motivated to incorporate a compliance server to perform compliance verification to confirm that said software file complies with a predetermined set of rules in order to efficiently produce a useful, reliable computer system which may be delivered to businesses and individuals free from errors and ready to run (*see Amberg et al. – Column 1: 40-43*).

Also, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the software file transferred to a target information handling system after verification that the software file complies with a set of predetermined parameters in the system of Application '823, since the system of Application '823 is operable to transfer the software file to a target information handling system. One would have been motivated to have the software file transferred to a target information handling system after verification that the software file complies with a set of predetermined parameters to ensure that the software meets the compatibility requirements of the computer system and to minimize the occurrence of errors during the subsequent installation of the software file.

Furthermore, Claim 1 of Application '823 recites that the software file comprises a plurality of individual constituent program files and a plurality of installation parameters associated with the software file and that the repack and script regeneration server: a) disassembles the software file into a first plurality of individual constituent program files; b) generates customized installation scripts in accordance with the plurality of parameters associated with the software file; and c) removes predetermined individual constituent program files in accordance with a prune list, thereby, creating a second plurality of individual constituent program files. These limitations are broader than the same limitations recited in Claim 1 of the instant application. Thus, these limitations of Claim 1 of the instant application are anticipated by the same limitations of Claim 1 of Application '823. *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993). Therefore, the recited functional limitations of "a distribution server" and "a repack and script regeneration server" in Claim 1 of the instant application would be an

Art Unit: 2191

obvious variation of the recited limitation of “a distribution server” and “a repack and script regeneration server” in Claim 1 of Application ‘823.

Copending Application 10/768,823	Instant Application 10/657,989
1. A system for automated dissemination of software to an information handling system, comprising:	1. A system for automated dissemination of software to an information handling system, comprising:
a distribution server operable to receive a software file <b><u>comprising a plurality of individual constituent program files and a plurality of installation parameters associated with said software file;</u></b>	a distribution server operable to receive a software file;
a repack and script regeneration server operably connected to said distribution server, wherein said repack and script regeneration server: a) disassembles said software file <b><u>into a first plurality of individual constituent program files;</u></b> b) <b><u>generates customized installation scripts in accordance with said plurality of installation parameters associated with said software file;</u></b> c) <b><u>removes predetermined individual constituent program files from said first plurality of individual constituent program files in accordance with a prune list, thereby creating a second plurality of individual constituent program files;</u></b> and d) repackages said second plurality of files combined with said customized installation scripts to provide automatic transfer of said software file to an information handling system; and	a repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software file;
	<b><u>a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system;</u></b>
	<b><u>a compliance server operably coupled to said distribution server, said compliance</u></b>



Art Unit: 2191

	<u>server being operable to perform compliance verification to confirm that said software file complies with a predetermined set of rules; and</u>
a download server operable to transfer said second plurality of files to a target information handling system.	a download server operable to transfer said software file to a target information handling system <u>after verifying that said software file complies with a predetermined set of rules.</u>

Claim 8 of the instant application is an independent method claim that corresponds to Claim 1 of the instant application, and as such is provisionally rejected for the reasons set forth in the provisional double patenting rejection of Claim 1 above.

Copending Application 10/768,823	Instant Application 10/657,989
9. A method for automated dissemination of software to an information handling system, comprising:	8. A method for automated dissemination of software to an information handling system, comprising:
receiving a software file <u>comprising a plurality of individual constituent program files and a plurality of installation parameters associated with said software file;</u>	receiving a software file;
disassembling said software file <u>into a plurality of individual constituent program files;</u>	disassembling said software file and
<u>generating customized installation scripts in accordance with said plurality of installation parameters associate with said software file;</u>	
<u>removing predetermined individual constituent program files from said first plurality of individual constituent program files in accordance with a prune list, thereby creating a second plurality of individual constituent program files;</u>	
repackaging said second plurality of files combined with said customized installation scripts to provide automatic transfer of said software file to an information handling system; and	repackaging said software file with scripts for automatically controlling the transfer of said software file;

	<u>generating commands to control the automatic downloading of software images of said software file to a target information handling system;</u>
	<u>using a compliance server to perform compliance verification to confirm that said software file complies with a predetermined set of rules; and</u>
transferring said second plurality of files to a target information handling system.	transferring said software file to a target information handling system <u>after verifying that said software file complies with a predetermined set of rules.</u>

Claim 15 of the instant application is an independent system claim that corresponds to Claim 1 of the instant application, and as such is provisionally rejected for the reasons set forth in the provisional double patenting rejection of Claim 1 above.

Copending Application 10/768,823	Instant Application 10/657,989
17. An information handling system, comprising:	15. An information handling system, comprising:
a data processor; and	a data processor; and
a data storage having a software file stored thereon, wherein said software is transferred to said data storage by an automated software dissemination system comprising:	data storage having a software file stored thereon, said software file being transferred to said data storage by an automated software dissemination system comprising:
a distribution server operable to receive a software file <u>comprising a plurality of individual constituent program files and a plurality of installation parameters associated with said software file;</u>	a distribution server operable to receive a software file;
a repack and script regeneration server operably connected to said distribution server, wherein said repack and script regeneration server: a) disassembles said software file <u>into a first plurality of individual constituent program files; b) generates customized installation scripts in accordance with said plurality of installation parameters associated with said software file; c)</u>	a repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software file;

Art Unit: 2191

<u>removes predetermined individual constituent program files from said first plurality of individual constituent program files in accordance with a prune list, thereby creating a second plurality of individual constituent program files;</u> and d) repackages said second plurality of files combined with said customized installation scripts to provide automatic transfer of said software file to an information handling system; and	
	<u>a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system;</u>
	<u>a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file compiles with a predetermined set of rules; and</u>
a download server operable to transfer said second plurality of files to a target information handling system.	a download server operable to transfer said software to said information handling system <u>after verification that said software file complies with a set of predetermined parameters.</u>

As per Claims 3, 5-7, 10, 12-14, 17, 19, and 20 of the instant application, the limitations in each of these dependent claims are also recited in Claims 5-8, 13-16, and 21-24, respectively, of Application '823.

This is a provisional obviousness-type double patenting rejection.

***Claim Rejections - 35 USC § 103***

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

16. **Claims 1, 3, 5, 6, 8, 10, 12, 13, 15, 17, 19, and 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Amberg et al.** (US 5,991,543) in view of **Feinman** (US 6,075,943).

As per **Claim 1**, **Amberg et al.** disclose:

- a distribution server operable to receive a software file (*see Figure 1: 140; Column 3: 48-55, "... sequencing program 204 residing on step maker 140, first reads a plurality of component descriptors from descriptor file 96." and 59-66, "... the component descriptors are included in a descriptor file called a system descriptor record which is a computer readable file containing a listing of the components, hardware and/or software components, to be installed onto target system 160."*);
- a repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1: 140; Column 4: 8-14, "Having retrieved the software installation and/or testing steps appropriate for target system 160, sequencing program 204 sequences the steps in a predetermined order according to sequence numbers corresponding to each step.*

Art Unit: 2191

*Having sequenced the steps required for target system 160, sequencing program 204 writes a series of output files to step disk 150.”; Column 9: 56-67 through Column 10: 1-2, “The result of the joinder of the Sys\_Comp table 112 and the Comp\_Step table 114 is then joined with the Sys\_Step\_Seq table 106 which contains all the steps for family X.” and “... a three-table join of Sys\_Comp table 112, Comp\_Step table 114, and Sys\_Step\_Seq table 106 yields the appropriate software installation and testing steps as well as sequencing information in the form of sequence and phase numbers to install and/or test software upon target computer system 160.”);*

- a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file complies with a predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, “... the sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108.”); and*

- a download server operable to transfer said software file to a target information handling system after verifying that said software file complies with a predetermined set of rules (*see Figure 1: 170 and 190; Figure 2: 202; Column 10: 61-64, “... the output files reside upon the server 202 or file server 190, where they can be used to direct the execution of the software installation and testing steps upon target computer system 160.”).*

However, Amberg et al. do not disclose:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands

to automatically control the downloading of software images of said software file to a target information handling system.

Feinman discloses:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system (*see Figure 1B: 11; Figure 7: 100; Column 3: 3-5, "... the server 11 shown has application programs which require automatic delivery to the remote clients 13." and 44-67 through Column 4: 1-4, "Once the application program(s) have been packed up, the next step is to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point and the time that each application program is to be delivered (FIG. 3)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to send the application program to the remote site at a specified time, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Feinman – Column 1: 23-32*).

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Amberg et al. further disclose:

- wherein said compliance server is operable to automatically generate a non-compliance notice message upon detection that said software file does not comply with said predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, "If a component is not legal (i.e. if a component in the system descriptor record is not contained in Component table 108), an error flag is raised."*).

As per **Claim 5**, the rejection of **Claim 1** is incorporated; and Amberg et al. further disclose:

- a test control server operable to verify proper operation of said software file on said target information handling system (*see Figure 2: 202; Figure 11; Column 12: 1-3, "... this program is called Runstep and is located on step disk 150 in the embodiment of FIG. 1 and on file server 202 in the embodiment of FIG. 2." and 62-67 through Column 13: 1-12, "... the Runstep program fist checks to see if a file named Re\_Run.bat exists. A Re\_Run.bat file is created before any command is executed from a step sequence and is removed after successful completion of the command. The existence of Re\_Run.bat indicates to the Runstep program in module 900 that the last command run was not successfully completed."*).

However, Amberg et al. do not disclose:

- a test control server operable to confirm the download of said software file to said target information handling system.

Feinman discloses:

- a test control server operable to confirm the download of said software file to said target information handling system (*see Figure 5: 70; Column 3: 40-43, "The list will be used later on by the automatic installation system to ensure that all files for an installation were transferred and unpacked successfully."*; Column 5: 18-21, "If the file transfer fails in step 70, an error code is trapped and an error message is placed into two of three logs maintained by the automatic installation system as shown in step 74.").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include a test control server operable to confirm the download of said software file to said target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to validate that the application delivery was successful, so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Feinman – Column 1: 33-37*).

As per **Claim 6**, the rejection of **Claim 1** is incorporated; and Amberg et al. further disclose:

- wherein said distribution server is operable to notify a manager regarding the status of said software file within a software distribution system (*see Column 14: 20-25, "... results from the software installation and testing may be logged back to either file server 190 or to file server 202. The results preferably include whether all the steps were completed successfully and what types of failures (if any) were encountered."*).



As per **Claim 8**, Amberg et al. disclose:

- receiving a software file (see Figure 1: 140; Column 3: 48-55, "... sequencing program 204 residing on step maker 140, first reads a plurality of component descriptors from descriptor file 96." and 59-66, "... the component descriptors are included in a descriptor file called a system descriptor record which is a computer readable file containing a listing of the components, hardware and/or software components, to be installed onto target system 160.");
- disassembling said software file and repackaging said software file with scripts for automatically controlling the transfer of said software file (see Figure 1: 140; Column 4: 8-14, "Having retrieved the software installation and/or testing steps appropriate for target system 160, sequencing program 204 sequences the steps in a predetermined order according to sequence numbers corresponding to each step. Having sequenced the steps required for target system 160, sequencing program 204 writes a series of output files to step disk 150."; Column 9: 56-67 through Column 10: 1-2, "The result of the joiner of the Sys\_Comp table 112 and the Comp\_Step table 114 is then joined with the Sys\_Step\_Seq table 106 which contains all the steps for family X." and "... a three-table join of Sys\_Comp table 112, Comp\_Step table 114, and Sys\_Step\_Seq table 106 yields the appropriate software installation and testing steps as well as sequencing information in the form of sequence and phase numbers to install and/or test software upon target computer system 160.");
- using a compliance server to perform compliance verification to confirm that said software file complies with a predetermined set of rules (see Figure 1: 100; Column 9: 9-16, "... the sequencing program and database determine if the processor, hard drive, monitor, and

*software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108.”); and*

- transferring said software file to a target information handling system after verifying that said software file complies with a predetermined set of rules (*see Figure 1: 170 and 190; Figure 2: 202; Column 10: 61-64, “... the output files reside upon the server 202 or file server 190, where they can be used to direct the execution of the software installation and testing steps upon target computer system 160.”*).

However, Amberg et al. do not disclose:

- generating commands to control the automatic downloading of software images of said software file to a target information handling system.

Feinman discloses:

- generating commands to control the automatic downloading of software images of said software file to a target information handling system (*see Figure 1B: 11; Figure 7: 100; Column 3: 3-5, “... the server 11 shown has application programs which require automatic delivery to the remote clients 13.” and 44-67 through Column 4: 1-4, “Once the application program(s) have been packed up, the next step is to identify the remote client’s delivery points, the application programs that are to be delivered to each delivery point and the time that each application program is to be delivered (FIG. 3).”*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include generating commands to control the automatic downloading of software images of said software file to a target information handling system. The modification would be obvious

because one of ordinary skill in the art would be motivated to send the application program to the remote site at a specified time, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Feinman – Column 1: 23-32*).

As per **Claim 10**, the rejection of **Claim 8** is incorporated; and Amberg et al. further disclose:

- generating a non-compliance notice message upon detection that said software file does not comply with said predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, “If a component is not legal (i.e. if a component in the system descriptor record is not contained in Component table 108), an error flag is raised.”*).

As per **Claim 12**, the rejection of **Claim 8** is incorporated; and Amberg et al. further disclose:

- confirming proper operation of said software file on said target information handling system (*see Figure 2: 202; Figure 11; Column 12: 1-3, “... this program is called Runstep and is located on step disk 150 in the embodiment of FIG. 1 and on file server 202 in the embodiment of FIG. 2.” and 62-67 through Column 13: 1-12, “... the Runstep program first checks to see if a file named Re\_Run.bat exists. A Re\_Run.bat file is created before any command is executed from a step sequence and is removed after successful completion of the command. The existence of Re\_Run.bat indicates to the Runstep program in module 900 that the last command run was not successfully completed.”*).

However, Amberg et al. do not disclose:

- confirming the download of said software file to said target information handling system.

Feinman discloses:

- confirming the download of said software file to said target information handling system (*see Figure 5: 70; Column 3: 40-43, "The list will be used later on by the automatic installation system to ensure that all files for an installation were transferred and unpacked successfully."; Column 5: 18-21, "If the file transfer fails in step 70, an error code is trapped and an error message is placed into two of three logs maintained by the automatic installation system as shown in step 74."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include confirming the download of said software file to said target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to validate that the application delivery was successful, so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Feinman – Column 1: 33-37*).

As per **Claim 13**, the rejection of **Claim 8** is incorporated; and Amberg et al. further disclose:

- notifying a manager regarding the status of said software file within a software distribution system (*see Column 14: 20-25, "... results from the software installation and testing*

*may be logged back to either file server 190 or to file server 202. The results preferably include whether all the steps were completed successfully and what types of failures (if any) were encountered.”).*

As per **Claim 15**, Amberg et al. disclose:

- a data processor (*see Column 3: 32-37, “Target system 160 might include ... a certain brand of processor ... ”*); and
- a data storage having a software file stored thereon (*see Column 3: 32-37, “Target system 160 might include a certain brand of hard drive ... ”*), said software file being transferred to said data storage by an automated software dissemination system comprising:
  - a distribution server operable to receive a software file (*see Figure 1: 140; Column 3: 48-55, “... sequencing program 204 residing on step maker 140, first reads a plurality of component descriptors from descriptor file 96.” and 59-66, “... the component descriptors are included in a descriptor file called a system descriptor record which is a computer readable file containing a listing of the components, hardware and/or software components, to be installed onto target system 160.”*);
  - a repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repack said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1: 140; Column 4: 8-14, “Having retrieved the software installation and/or testing steps appropriate for target system 160, sequencing program 204 sequences the steps in a predetermined order according to sequence numbers corresponding to each step.*

Art Unit: 2191

*Having sequenced the steps required for target system 160, sequencing program 204 writes a series of output files to step disk 150.”; Column 9: 56-67 through Column 10: 1-2, “The result of the joinder of the Sys\_Comp table 112 and the Comp\_Step table 114 is then joined with the Sys\_Step\_Seq table 106 which contains all the steps for family X.” and “... a three-table join of Sys\_Comp table 112, Comp\_Step table 114, and Sys\_Step\_Seq table 106 yields the appropriate software installation and testing steps as well as sequencing information in the form of sequence and phase numbers to install and/or test software upon target computer system 160.”);*

- a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file complies with a predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, “... the sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108.”); and*

- a download server operable to transfer said software file to said information handling system after verifying that said software file complies with a predetermined set of rules (*see Figure 1: 170 and 190; Figure 2: 202; Column 10: 61-64, “... the output files reside upon the server 202 or file server 190, where they can be used to direct the execution of the software installation and testing steps upon target computer system 160.”).*

However, Amberg et al. do not disclose:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands

to automatically control the downloading of software images of said software file to said information handling system.

Feinman discloses:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system (*see Figure 1B: 11; Figure 7: 100; Column 3: 3-5, "... the server 11 shown has application programs which require automatic delivery to the remote clients 13." and 44-67 through Column 4: 1-4, "Once the application program(s) have been packed up, the next step is to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point and the time that each application program is to be delivered (FIG. 3)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to send the application program to the remote site at a specified time, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Feinman – Column 1: 23-32*).

As per **Claim 17**, the rejection of **Claim 15** is incorporated; and Amberg et al. further disclose:

- wherein said compliance server is operable to automatically generate a non-compliance notice message upon detection that said software file does not comply with said predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, "If a component is not legal (i.e. if a component in the system descriptor record is not contained in Component table 108), an error flag is raised."*).

As per **Claim 19**, the rejection of **Claim 15** is incorporated; and Amberg et al. further disclose:

- a test control server operable to verify proper operation of said software file on said target information handling system (*see Figure 2: 202; Figure 11; Column 12: 1-3, "... this program is called Runstep and is located on step disk 150 in the embodiment of FIG. 1 and on file server 202 in the embodiment of FIG. 2." and 62-67 through Column 13: 1-12, "... the Runstep program fist checks to see if a file named Re\_Run.bat exists. A Re\_Run.bat file is created before any command is executed from a step sequence and is removed after successful completion of the command. The existence of Re\_Run.bat indicates to the Runstep program in module 900 that the last command run was not successfully completed."*).

However, Amberg et al. do not disclose:

- a test control server operable to confirm the download of said software file to said target information handling system.



Feinman discloses:

- a test control server operable to confirm the download of said software file to said target information handling system (*see Figure 5: 70; Column 3: 40-43, "The list will be used later on by the automatic installation system to ensure that all files for an installation were transferred and unpacked successfully."; Column 5: 18-21, "If the file transfer fails in step 70, an error code is trapped and an error message is placed into two of three logs maintained by the automatic installation system as shown in step 74."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include a test control server operable to confirm the download of said software file to said target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to validate that the application delivery was successful, so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Feinman – Column 1: 33-37*).

As per **Claim 20**, the rejection of **Claim 15** is incorporated; and Amberg et al. further disclose:

- wherein said distribution server is operable to notify a manager regarding the status of said software file within a software distribution system (*see Column 14: 20-25, "... results from the software installation and testing may be logged back to either file server 190 or to file server 202. The results preferably include whether all the steps were completed successfully and what types of failures (if any) were encountered."*).

17. **Claims 4, 11, and 18** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Amberg et al.** (US 5,991,543) in view of **Feinman** (US 6,075,943) as applied to Claims 1, 8, and 15 above, and further in view of **Tso et al.** (US 6,088,803).

As per **Claim 4**, the rejection of **Claim 1** is incorporated; however, **Amberg et al.** and **Feinman** do not disclose:

- wherein said distribution server is operable to scan said software file for viruses.

**Tso et al.** disclose:

- wherein said distribution server is operable to scan said software file for viruses (*see Figure 2: 40; Column 2: 62-67 through Column 3: 1-5, "... network device 4 invokes virus checker 5, which in turn performs its preconfigured virus scan processing with the requested file as input (Step 40)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of **Tso et al.** into the teaching of **Amberg et al.** to include wherein said distribution server is operable to scan said software file for viruses. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize breaches in system integrity (*see Tso et al. – Column 1: 27-28*).

As per **Claim 11**, the rejection of **Claim 8** is incorporated; however, **Amberg et al.** and **Feinman** do not disclose:

- scanning said software file for viruses.

Tso et al. disclose:

- scanning said software file for viruses (*see Figure 2: 40; Column 2: 62-67 through Column 3: 1-5, "... network device 4 invokes virus checker 5, which in turn performs its preconfigured virus scan processing with the requested file as input (Step 40)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to include scanning said software file for viruses. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize breaches in system integrity (*see Tso et al. – Column 1: 27-28*).

As per **Claim 18**, the rejection of **Claim 15** is incorporated; however, Amberg et al. and Feinman do not disclose:

- wherein said distribution server is operable to scan said software file for viruses.

Tso et al. disclose:

- wherein said distribution server is operable to scan said software file for viruses (*see Figure 2: 40; Column 2: 62-67 through Column 3: 1-5, "... network device 4 invokes virus checker 5, which in turn performs its preconfigured virus scan processing with the requested file as input (Step 40)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to include wherein said distribution server is operable to scan said software file for viruses. The

Art Unit: 2191

modification would be obvious because one of ordinary skill in the art would be motivated to minimize breaches in system integrity (*see Tso et al.* – Column 1: 27-28).

18. **Claims 7 and 14** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Amberg et al.** (US 5,991,543) in view of **Feinman** (US 6,075,943) as applied to Claims 1 and 8 above, and further in view of **Karasudani et al.** (US 6,378,054).

As per **Claim 7**, the rejection of **Claim 1** is incorporated; however, **Amberg et al.** and **Feinman** do not disclose:

- an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon.

**Karasudani et al.** disclose:

- an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon (*see Column 11: 60-61, "When the m<sup>th</sup> data file is added to the archive database ... "*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of **Karasudani et al.** into the teaching of **Amberg et al.** to include an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize damages by immediately recovering a data file in the event of problems, such as the loss of data (*see Karasudani et al.* – Column 1: 26-29).

As per **Claim 14**, the rejection of **Claim 8** is incorporated; however, Amberg et al. and Feinman do not disclose:

- transferring copies of said software file to an archive server for storage thereon.

Karasudani et al. disclose:

- transferring copies of said software file to an archive server for storage thereon (*see Column 11: 60-61, "When the m<sup>th</sup> data file is added to the archive database ..."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Karasudani et al. into the teaching of Amberg et al. to include transferring copies of said software file to an archive server for storage thereon. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize damages by immediately recovering a data file in the event of problems, such as the loss of data (*see Karasudani et al. – Column 1: 26-29*).

### ***Response to Arguments***

19. Applicant's arguments filed on February 6, 2007 have been fully considered, but they are not persuasive.

***In the remarks, Applicant argues that:***

- a) Independent claims 1, 8 and 15 have been amended to add the limitation of a compliance server operably connected to the distribution server. The compliance server is operable to perform compliance verification to confirm that the "software file complies with a predetermined

set of rules." This limitation was previously recited in claims 2, 9, and 16, which Examiner rejected, based on the disclosure in column 9, lines 9-1.6 of the Amberg reference. The cited portion of the Amberg reference describes a system wherein various components are compared to "component table" prior to installation on an information handling system. The cited portion of Amberg does not provide a teaching of a compliance server that is operable to perform compliance verification to confirm that a software file complies with a predetermined set of rules, as recited in amended independent claims 1, 8 and 15.

***Examiner's response:***

a) Examiner disagrees. Amberg et al. clearly disclose "a compliance server that is operable to perform compliance verification to confirm that a software file complies with a predetermined set of rules" (*see Figure 1: 100; Column 9: 9-16, "... the sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108."*). Note that the entries in the component table are interpreted as the predetermined rules, where the component table is used to check if the components (processor, hard drive, monitor, and software) of the target computer system are legal (*i.e.*, included on a target computer system).

***Conclusion***

20. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Art Unit: 2191

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / QC  
March 15, 2007



WEI ZHEN  
SUPERVISORY PATENT EXAMINER